



API Reference Guide

B-gate Xamarin SDK

Rev. 0.01

Introduction

The B-gate series printers are designed to be used by connecting them to other electronic products such as ECR (Electronic Cash Registers), POS (Point Of Sales), and computer peripheral devices.

For those who have newly purchased a printer, please read these instructions carefully before using the product.

We at BIXOLON maintain ongoing efforts to enhance and upgrade the functions and quality of all our products. In following, product specifications and/or user manual content may be changed without prior notice.

Table of Contents

1. Manual Guide	5
2. Xamarin SDK Installation	5
2-1 Supported Platforms	5
2-1-1 iOS.....	5
2-1-2 Android.....	5
2-1-3 Windows.....	6
2-2 Supported Communications	6
2-2-1 Bluetooth	6
2-2-2 WiFi / Ethernet	6
2-3 Installation Instruction	7
2-3-1 Installation steps from NuGet server	8
2-3-2 Installation steps from a local nupkg file	8
3. IMPosDeviceFactory Reference	11
3-1 Overview	11
3-2 Methods	11
3-2-1 createDevice	11
4. MPosControllerDevices Reference.....	12
4-1 Overview	12
4-2 Properties	12
4-2-1 dataEvent.....	12
4-2-2 statusUpdateEvent	12
4-2-3 DeviceID.....	12
4-2-3 IsOpen.....	12
4-2-3 CommandMode	12
4-2-4 InterfaceType	12
4-3 Methods	13
4-3-1 selectInterface	13
4-3-2 selectCommandMode	14
4-3-3 openService	14
4-3-4 closeService.....	14
4-3-5 directIO	15
4-3-6 setTransaction	16
5. MPosControllerPrinter Reference.....	17
5-1 Overview	17
5-2 Methods	17
5-2-1 printText	17
5-2-2 setCharacterSet	18
5-2-3 setInternationalCharacterSet	18
5-2-4 setPagemode.....	18
5-2-5 setPagemodePrintArea	19
5-2-6 setPagemodeDirection	20
5-2-7 setPagemodePosition	20
5-2-8 printBitmapFile.....	21
5-2-9 print1DBarcode.....	22
5-2-10 printQRCode.....	23

Xamarin SDK API Reference Guide

5-2-11 printPDF417	24
5-2-12 checkPrinterStatus	25
6. MPosLookup Reference	26
6-1 Overview	26
6-2 Methods	26
6-2-1 refreshDevicesList	26
6-2-2 getDevicesList	26

1. Manual Guide

- This SDK manual provides descriptions of contents necessary for developing Xamarin.Forms PCL (Portable Class Library) applications. Because Xamarin API library and Demo application were built using Visual Studio 2015, all the contents in this manual are explained based on Visual Studio 2015.

2. Xamarin SDK Installation

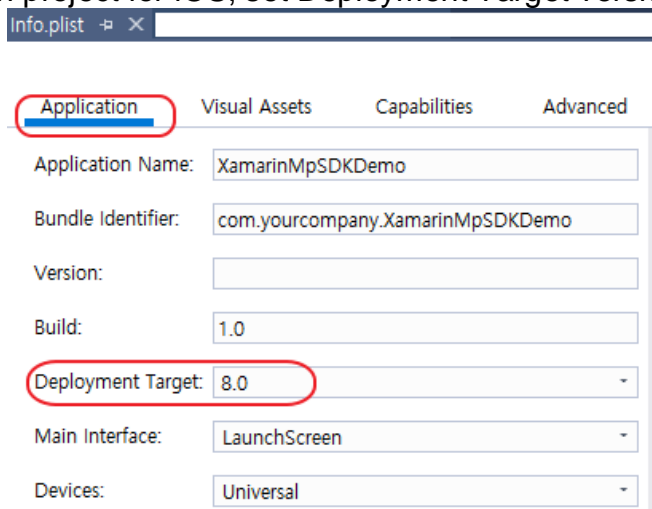
2-1 Supported Platforms

2-1-1 iOS

- **iOS 8.0 or later**

Note

In your Xamarin project for iOS, set Deployment Target version to 8.0.



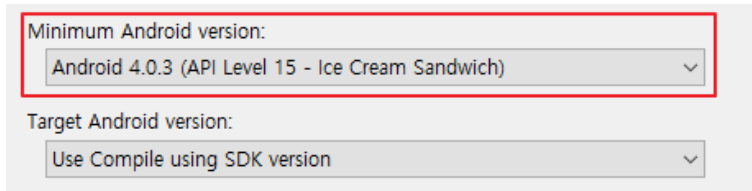
The screenshot shows the 'Info.plist' file in a text editor. The 'Application' tab is selected. The 'Deployment Target' is set to '8.0'. Other fields include 'Application Name' (XamarinMpSDKDemo), 'Bundle Identifier' (com.yourcompany.XamarinMpSDKDemo), 'Version' (empty), 'Build' (1.0), 'Main Interface' (LaunchScreen), and 'Devices' (Universal).

2-1-2 Android

- **Android 4.0.3(API Level 15 : ice cream sandwich) or later**

Note

In your Xamarin project for android, set minimum android version to API Level 15.



The screenshot shows the 'Minimum Android version' dropdown menu set to 'Android 4.0.3 (API Level 15 - Ice Cream Sandwich)'. The 'Target Android version' dropdown menu is set to 'Use Compile using SDK version'.

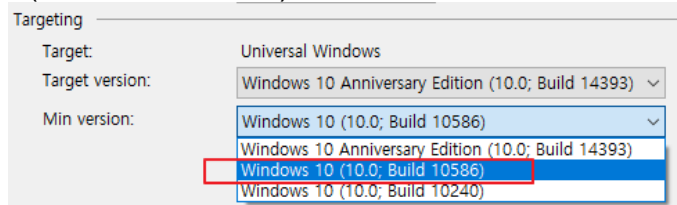
You should leave the target Android version set to Use Compile using SDK version (as shown) so that its value automatically matches the target framework setting. In general, the Target Android Version should be bounded by the Minimum Android Version and the Target Framework. That is: **Minimum Android Version < Target Android Version < Target Framework**.

2-1-3 Windows

- **Windows 10 version Build 10586 or later.**

Note

In your Xamarin project for Universal Windows, set minimum version to Windows 10 (10.0 Build 10586) as below.



2-2 Supported Communications

2-2-1 Bluetooth

- HOST must be paired with Bluetooth device for Bluetooth communication in advance. Make sure to include necessary permissions (or capabilities) to each platform before using APIs in this manual. Please refer to “2-3” in this manual for permission settings.

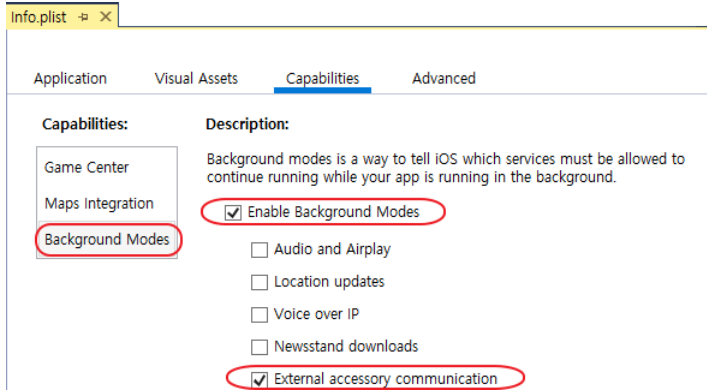
2-2-2 WiFi / Ethernet

- HOST in the same network to the devices can be communicated with TCP/IP over wired or wireless network. Make sure to include necessary permissions (or capabilities) to each platform before using APIs in this manual. Please refer to “2-3” in this manual for permission settings.

2-3 Permission / Capabilities

When developing application using Bluetooth and Wifi, there are a few things to modify your project's settings and configuration information related to permissions for each platform. Demo application provided helps you modifying the necessary settings of your own Xamarin application.

Platform	Required permission / capabilities
iOS	<p>1. Update the Info.plist file by opening it in XML editor or text editor. Add the following to the file. This is to allow your app to access the Bluetooth port.</p> <div><pre><key>UISupportedExternalAccessoryProtocols</key> <array><string>com.bixolon.protocol</string></array></pre></div>

	<p>2. Put a checkmark on “External Accessory communication” checkbox as a below.</p> 
Android	<p>Make sure to include following capabilities in the AndroidManifest.xml file.</p> <ul style="list-style-type: none"> - BLUETOOTH, BLUETOOTH_ADMIN, BLUETOOTH_PRIVILEGED - ACCESS_WIFI_STATE, CHANGE_WIFI_STATE <pre><manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="auto"> <uses-sdk android:minSdkVersion="15" /> <uses-permission android:name="android.permission.BLUETOOTH" /> <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" /> <uses-permission android:name="android.permission.BLUETOOTH_PRIVILEGED" /> <uses-permission android:name="android.permission.ACCESS_WIFI_STATE" /> <uses-permission android:name="android.permission.CHANGE_WIFI_STATE" /> </manifest></pre>
Windows	<p>Make sure to include following capabilities in the “Package.appxmanifest” file.</p> <ul style="list-style-type: none"> - Bluetooth, Internet (Client), Private Networks (Client & Server)

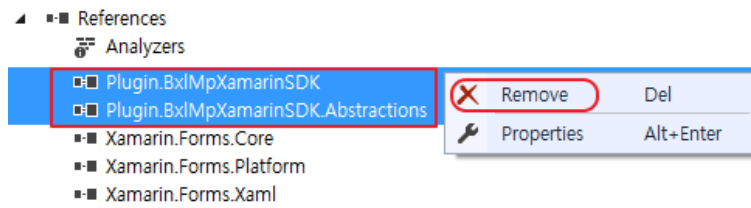
2-3 Installation Instruction

Xamarin SDK can be installed as a **NuGet** package into MS Visual Studio. When you use NuGet to install a package, it copies the library files to your solution and automatically updates your project (add references, change config files, etc). If you remove a package, **NuGet** reverses whatever changes it made so that no clutter is left. If you need more information about NuGet, visit <https://www.nuget.org/>

Xamarin SDK API Reference Guide

Note

Before installing NuGet Package for Bixelon Xamarin SDK, library references below must be removed if you added them manually on each platform.



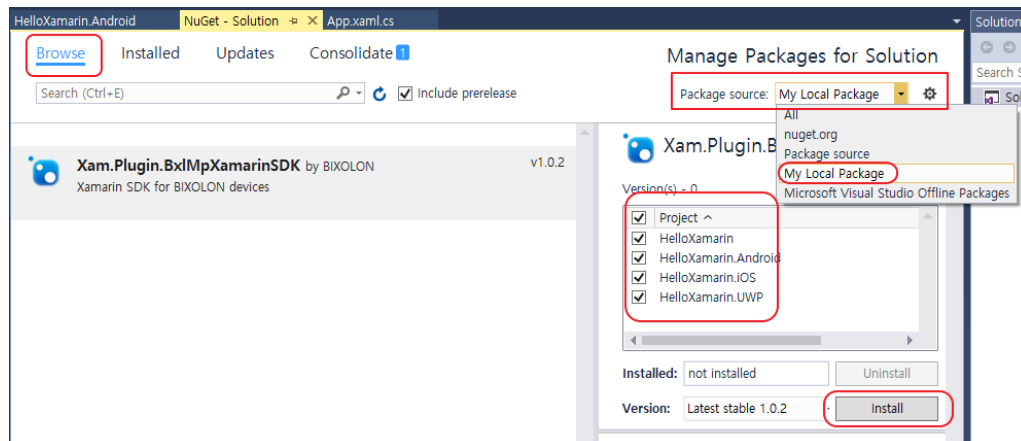
2-3-1 Installation steps from NuGet server

- This way isn't supported yet. Refer to "2-4-2" below in this manual to install Xamarin SDK.

2-3-2 Installation steps from a local nupkg file

#	Description
1	<p>Open Visual Studio, and open your own Xamarin.Forms solution.</p> <p>The screenshot shows the 'File' menu in Visual Studio. The 'Open' option is highlighted with a red box. A sub-menu is open, showing 'Project/Solution...' (Ctrl+Shift+O) also highlighted with a red box. Other options in the sub-menu include 'Web Site...', 'Open from Source Control', 'Team Project...', 'File...' (Ctrl+O), and 'Convert...'.</p>
2	<p>From the Solution Explorer, right click the top level solution.</p> <p>The screenshot shows the 'Solution Explorer' in Visual Studio. The top-level solution 'Solution 'HelloXamarin' (4 projects)' is highlighted with a blue background and a red circle. Below it, four projects are listed: 'HelloXamarin (Portable)', 'HelloXamarin.Android', 'HelloXamarin.iOS', and 'HelloXamarin.UWP (Universal Windows)'.</p>
3	<p>Click "Manage NuGet Packages for Solution..."</p> <p>The screenshot shows a context menu with several options. The option 'Manage NuGet Packages for Solution...' is highlighted with a red box. Other options include 'Configuration Manager...', 'Archive All...', 'View Archives', and 'Restore NuGet Packages'.</p>
4	<p>On the top of the right side, click "Setting" button in red.</p>

Xamarin SDK API Reference Guide



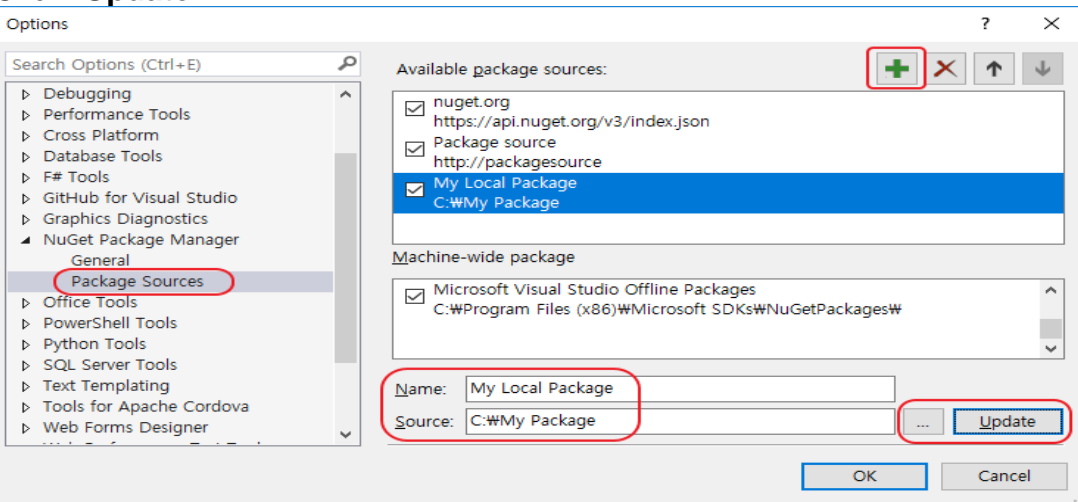
5 Click **“Package Sources”**, and then click **“+”** button.

Browse to the directory where **“Xam.Plugin.BxlMpXamarinSDK.x.x.x.nupkg”** file exists, and then enter any name you want.

6

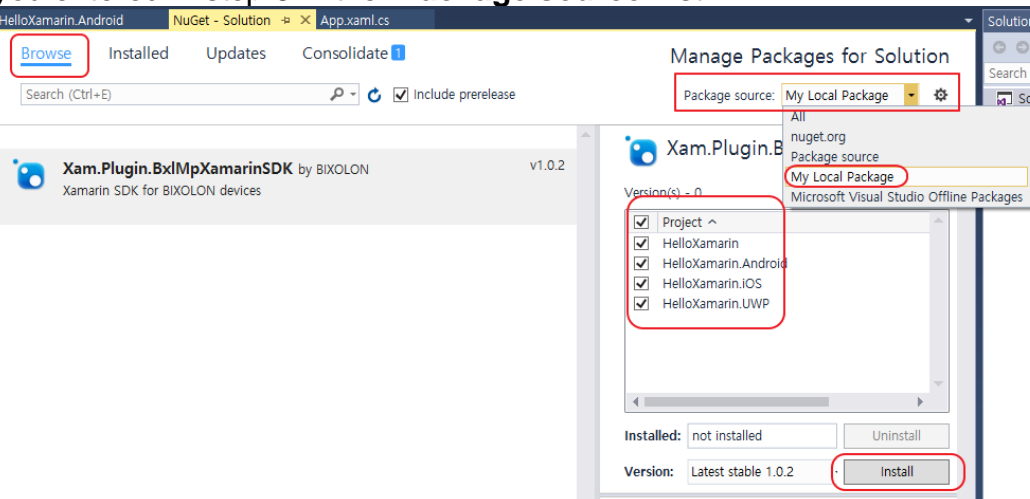
Note “x.x.x” is the version number for the SDK.

Click **“Update”**


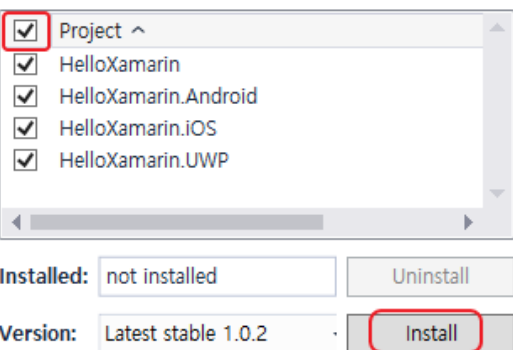


8 Click **“OK”**.

“Xam.Plugin.BxlMpXamarinSDK” package can be found when you select the name you entered in step 6 in the **“Package source”** list.



9

10	<p>Click “Project” checkbox to install the entire package.</p> <p> Xam.Plugin.BxlMpXamarinSDK</p> <p>Version(s) - 0</p>  <p>Installed: not installed Uninstall</p> <p>Version: Latest stable 1.0.2 Install</p>
11	<p>Click “Install” to start installing all the appropriate references to the libraries for each platform project.</p>

3. IMPosDeviceFactory Reference

3-1 Overview

• "IMPosDeviceFactory" interface can be used to get an instance implementing "MPosControllerDevices" interface for communication with a specific device. All the instances related to the communication with devices are created using "createDevice" method.

3-2 Methods

Note

Use the "current" property in "MPosDeviceFactory" class to call the methods defined in this interface.

eg) MPosDeviceFactory.Current.createDevice(10)

3-2-1 createDevice

• Gets an instance for communication with a specific device.

[Syntax]

```
MPosControllerDevices createDevice(MPosDeviceType deviceType);
```

[Parameters]

MPosDevice deviceType: A value of the device type to be created.

MPosDeviceType Enumeration	Value	Description
MPOS_DEVICE_LABEL_PRINTER	1	Label Printer
MPOS_DEVICE_PRINTER	10	POS Printer
MPOS_DEVICE_MSR	30	MSR
MPOS_DEVICE_SCANNER	40	Barcode Scanner
MPOS_DEVICE_RFID	60	RFID
MPOS_DEVICE_DALLAS_KEY	70	Dallas Key
MPOS_DEVICE_NFC	80	NFC
MPOS_DEVICE_DISPLAY	110	Customer Display
MPOS_DEVICE_USB_TO_SERIAL	120	USB-to-Serial
MPOS_DEVICE_SCALE	130	Scale

[Returns]

A reference to the instance implementing "MPosControllerDevices" interface specified or null.

4. MPosControllerDevices Reference

4-1 Overview

• “**MPosControllerDevices**” interface provides common member methods with regards to the device, and the properties and methods can be used in all the classes implementing this interface.

4-2 Properties

4-2-1 dataEvent

• If data entry occurs in a device, the data in byte array will be sent to the application through this event.

4-2-2 statusUpdateEvent

• If the status of a printer device changes, the new status will be sent to the application through the this event. The sent value is an integer type constant value that has been defined in advance.

MPosPrinterStatus Enumeration	Value	Description
MPOS_STATUS_NORMAL	0	Printing is possible
MPOS_STATUS_PRINTER_COVEROPEN	1	Cover is open
MPOS_STATUS_PRINTER_PAPEREMPTY	2	No paper
MPOS_STATUS_PRINTER_PAPER_NEAREND	4	Insufficient paper
MPOS_STATUS_PRINTER_ERROR	8	Error
MPOS_STATUS_BUSY	1200	Communicating with another host
MPOS_STATUS_PROGRESS	1201	Transmitting
MPOS_STATUS_COMPLETED	1202	Transmission complete
MPOS_STATUS_REQUIRE_RESET	1203	B-gate reset required
MPOS_STATUS_FAIL	1204	Transmission failed

4-2-3 DeviceID

• The device Id of the device.

4-2-3 IsOpen

• The device in the usable status or not.

4-2-3 CommandMode

• Communication way to the device via “B-gate” or directly connection.

4-2-4 InterfaceType

• Communication Interface such as Bluetooth, Wireless LAN or Wired Lan.

4-3 Methods

4-3-1 selectInterface

- Configures the interface type and address you wish to connect. This method must be used before calling the “**openService**” method.

[Syntax]

int selectInterface(int interfaceType, string address);

[Parameters]

- int interface type: communication type to use.

MPosInterfaceType Enumeration	Value	Description
MPOS_INTERFACE_WIFI	1	Use when connecting to Wifi.
MPOS_INTERFACE_ETHERNET	2	Use when connecting to Ethernet.
MPOS_INTERFACE_BLUETOOTH	4	Use when connecting to Bluetooth.

- string address:

For TCP, use the “IP address [:tcp port number]” format. The tcp port number can be omitted, and the default value is 9100. If the tcp port number of B-gate has been changed, the new number must be used.

For a Bluetooth interface, the appropriate value depending on the plaforms must be entered.

Platform	Address value
iOS	Bluetooth Serial Number
Android	Bluetooth MAC Address
Windows	Bluetooth MAC Address

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_OPENED	1010	Called after openService method
MPOS_FAIL_INVALID_PARAMETER	1004	Parameter is invalid
MPOS_FAIL	1000	Failed

4-3-2 **selectCommandMode**

- Configures whether the host and the relevant device will communicate directly or through “B-gate”. This function must be used before the calling the “**openService**” function. If this method is not used, it will be configured as the default value.

[Syntax]

int selectCommandMode(int commandMode);

[Parameters]

- int commandMode: Select the mode to communicate with B-gate or a device.

MPosCommandMode Enumeration	Value	Description
MPOS_COMMAND_MODE_DEFAULT	0	Communicate through B-gate (default)
MPOS_COMMAND_MODE_BYPASS	1	Direct communication

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_OPENED	1010	Called after openService method
MPOS_FAIL_INVALID_PARAMETER	1004	Parameter is invalid
MPOS_FAIL	1000	Failed

4-3-3 **openService**

- In case of “bypass” mode by “**selectCommandMode**” method, “**openService**” method checks that the specified device by the address parameter of “**selectInterface**” method is ready for communication with HOST. In other case, “**openService**” method checks that the device with the smallest device Id value is ready for communication with HOST.

[Syntax]

Task<int> **openService**(uint timeout = 3);

[Parameters]

- uint timeout: Timeout value for connecting a device. Default value is 3 seconds.

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_ALREADY_OPEN	1	Already connected
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_NO_RESPONSE	1005	No response
MPOS_FAIL	1000	Failed

4-3-4 **closeService**

- Ends the use of a device.

[Syntax]

Task<int> closeService (int closeTimeout = 0);

[Parameters]

- int closeTimeout: Timeout value for disconnecting a device. Default value is 0 second.

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_ALREADY_OPEN	1	Already connected
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_NO_RESPONSE	1005	No response
MPOS_FAIL	1000	Failed

4-3-5 directIO

- Instantly sends data to the device.

[Syntax]

Task<int> directIO(byte[] data);

[Parameters]

- byte[] data: Data to be sent to device.

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NO_OPEN	1002	Device cannot be used
MPOS_FAIL	1000	Failed

4-3-6 **setTransaction**

- Configures the Transaction mode or cancels it after sending Transaction data to B-gate. Using the Transaction mode in a multi-host environment will guarantee data integrity between hosts. If another host is communicating with B-gate, the transmission and processing of sent data may be delayed, while the processing results of data in B-gate can be checked through StatusUpdateEvent.

Note

- It is recommended to use the Transaction mode in multi-host environments when using output devices such as printers, Customer displays, label printers, etc.

[Syntax]

Task<int> setTransaction(int mode);

[Parameters]

- int mode: Configure the transaction mode

MPosTransactionMode Enumeration	Value	Description
MPOS_PRINTER_TRANSACTION_OUT	0	Cancel transaction mode after data transmission
MPOS_PRINTER_TRANSACTION_IN	1	Enter transaction mode

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_TRANSACTION_ALREADY_IN	1015	Already in transaction mode
MPOS_FAIL	1000	Failed

5. MPosControllerPrinter Reference

5-1 Overview

- MPosControllerPrinter interface provides methods regarding Thermal Printer.

5-2 Methods

5-2-1 **printText**

- This is a method that prints characters. The character attributes be printed can be configured using "MPosFontAttribute".

[Syntax]

Task<int> printText(string data, MPosFontAttribute attribute);

[Parameters]

- string data: Character string to be printed
- MPosFontAttribute: Class Object for the font attributes.

```
class MPosFontAttribute
{
    public MPosAlignment Alignment { get; set; }
    public bool Bold { get; set; }
    public int CodePage { get; set; }
    public MPosFontType FontType { get; set; }
    public MPosFontSizeHeight Height { get; set; }
    public bool Reverse { get; set; }
    public MPosFontUnderline Underline { get; set; }
    public MPosFontSizeWidth Width { get; set; }
}
```

Properties

- Alignment: specifies characters alignments (Left, Center, Right)
- Bold: specifies whether the font is bold.
- CodePage: used for encoding characters.
- FontType: specifies one of the font types (Font A, Font B, Font C)
- Height: specifies the font height.
Note: after comparing to font width, the bigger size will be used as font size.
- Width: specifies the font width.
Note: after comparing to font height, the bigger size will be used as font size.
- Reverse: specifies whether the font is reversed.
- Underline: specifies whether the font is underlined.

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL	1000	Failed

5-2-2 setCharacterSet

- Configures the printer's codepage. The appropriate codepage value must be used in each printer. If the printer's codepage has already been configured with the printer configuration utility, this method does not need to be used.

[Syntax]

Task<int> setCharacterSet(int characterSet);

[Parameters]

- int characterSet: The printer's codepage value

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL	1000	Failed

5-2-3 setInternationalCharacterSet

- Configures the printer's international characterSet.

[Syntax]

Task<int> setInternationalCharacterSet(int internationalCharSet);

[Parameters]

- int internationalCharSet: The printer's international characterSet value

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL	1000	Failed

5-2-4 setPagemode

- Configures pagemode or cancels it after printing.

Note	<ul style="list-style-type: none">- Pagemode cannot be used simultaneously with transaction mode.- Printing with pagemode will guarantee data integrity in a multi-host environment.
-------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

[Syntax]

Task<int> setPagemode (int mode);

[Parameters]

- int mode: Configure pagemode. The defined constant values are listed below.

MPosPageMode Enumeration	Value	Description
MPOS_PRINTER_PAGEMODE_IN	1	Enter Pagemode
MPOS_PRINTER_PAGEMODE_OUT	0	Cancel Pagemode after printing

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_PAGEMODE_ALREADY_IN	1014	Already in pagemode
MPOS_FAIL	1000	Failed

5-2-5 setPagemodePrintArea

- Configures the printing area of pagemode.

[Syntax]

int setPagemodePrintArea(int x, int y, int width, int height);

[Parameters]

- int x: The starting x value of the pagemode printing area
- int y: The starting y value of the pagemode printing area
- int width: Width value of pagemode printing area
- int height: Height value of pagemode printing area

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL	1000	Failed

5-2-6 setPagemodeDirection

- Configures the printing direction in pagemode.

[Syntax]

```
int setPagemodeDirection(int direction);
```

[Parameters]

- int direction: Printing direction.

MPosPageMode Enumeration	Value	Description
MPOS_PRINTER_PD_0	0	0 degree turn
MPOS_PRINTER_PD_LEFT90	1	90 degree turn counter-clockwise
MPOS_PRINTER_PD_180	2	180 degree turn
MPOS_PRINTER_PD_RIGHT90	3	90 degree turn clockwise

[Returns]

MPosPageModeDirection Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL	1000	Failed

5-2-7 setPagemodePosition

- Configures the coordinates of the printing target in pagemode.

[Syntax]

```
int setPagemodePosition(int x, int y);
```

[Parameters]

- int x: x axis coordinate value of object to be printed
- int y: y axis coordinate value of object to be printed

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL	1000	Failed

5-2-8 printBitmapFile

- Prints an image file.

[Syntax]

Task<int> printBitmapFile(string path, long width, long alignment, long brightness,
bool isDithering, bool isCompress);

[Parameters]

- string path: The path of the image file to be printed.

Note In Android, the image file should be in the SD card, and used as this parameter.

- int width: The width of the printed image. Use a value larger than 0 or a constant value below.

MPosImageWidth Enumeration	Value	Description
MPOS_IMAGE_WIDTH_ASIS	-2	Print as original image size

- int alignment: Alignment.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align to the left
MPOS_ALIGNMENT_CENTER	1	Align at the center
MPOS_ALIGNMENT_RIGHT	2	Align to the right

- int brightness: The brightness of the image when printing. Can use a number between 0 and 100
- bool isDithering: Whether or not to apply dithering when printing image
- bool isCompress: Whether or not to compress when sending data

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL_NO_FILE	1000	There is no file
MPOS_FAIL	1000	Failed

5-2-9 **print1DBarcode**

- Prints a 1-D barcode.

[Syntax]

Task<int> print1DBarcode(string data, int symbology, int height, int barWidth, int alignment, int textPosition);

[Parameters]

- string data: Data of the barcode to be printed
- int symbology: The type of 1-D barcode.

MPos1dCodeType Enumeration	Value	Description
MPOS_BARCODE_UPCA	101	UPC A barcode
MPOS_BARCODE_UPCE	102	UPC E barcode
MPOS_BARCODE_EAN8 MPOS_BARCODE_JAN8	103	EAN 8/JAN 8 barcode
MPOS_BARCODE_EAN13 MPOS_BARCODE_JAN13	104	EAN 13/JAN 13 barcode
MPOS_BARCODE_ITF	106	ITF barcode
MPOS_BARCODE_CODABAR	107	CODABAR barcode
MPOS_BARCODE_CODE39	108	CODE39 barcode
MPOS_BARCODE_CODE93	109	CODE93 barcode
MPOS_BARCODE_CODE128	110	CODE128 barcode

- int height: Barcode height. Can use a number between 1 and 255
- int barWidth: Width of barcode bars. Can use a number between 2 and 8 (default value: 2)
- int alignment: Alignment value.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align left
MPOS_ALIGNMENT_CENTER	1	Align center
MPOS_ALIGNMENT_RIGHT	2	Align right

- int textPosition: The printing position of the HRI (Human Readable Interface).

MPosLabelHRI Enumeration	Value	Description
MPOS_BARCODE_TEXTNONE	0	Does not print
MPOS_BARCODE_TEXTABOVE	1	Print above barcode
MPOS_BARCODE_TEXTBELOW	2	Print below barcode

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL_2D_BARCODE	1016	Symbology is a 2-D barcode
MPOS_FAIL	1000	Failed

5-2-10 **printQRCode**

- Prints the QR Code.

[Syntax]

int printQRCode (string data, int model, int alignment, int moduleSize, int eccLevel);

[Parameters]

- string data: The QR Code data to be printed
- int model: The model type of the QR Code.

MPos2dCodeType Enumeration	Value	Description
MPOS_BARCODE_QRCODE_MODEL1	204	MODEL 1
MPOS_BARCODE_QRCODE_MODEL2	205	MODEL 2

- int alignment: Alignment value.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align left
MPOS_ALIGNMENT_CENTER	1	Align center
MPOS_ALIGNMENT_RIGHT	2	Align right

- int moduleSize: The module size of the QR Code. Can use a number between 1 and 8 (default value: 3)
- int eccLevel: Error correction level.

MPosQRCodeECCLevel Enumeration	Value	Description
MPOS_QRCODE_ECC_LEVEL_L	48	7% of codewords can be restored.
MPOS_QRCODE_ECC_LEVEL_M	49	15% of codewords can be restored.
MPOS_QRCODE_ECC_LEVEL_Q	50	25% of codewords can be restored.
MPOS_QRCODE_ECC_LEVEL_H	51	30% of codewords can be restored.

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL	1000	Failed

5-2-11 **printPDF417**

- Prints the PDF417 code.

[Syntax]

Task<int> **printPDF417**(string data, int symbol, int alignment, int columnNumber, int rowNumber, int moduleWidth, int moduleHeight, int eccLevel);

[Parameters]

- string data: The PDF417 code data to be printed
- int symbol: The type of the PDF417 code.

MPos2dCodeType Enumeration	Value	Description
MPOS_BARCODE_PDF417	201	PDF417 Standard
MPOS_BARCODE_PDF417_SIMPLIFIED	202	PDF417 Simplified

- int alignment: Alignment value.

MPosAlignment Enumeration	Value	Description
MPOS_ALIGNMENT_LEFT	0	Align left
MPOS_ALIGNMENT_CENTER	1	Align center
MPOS_ALIGNMENT_RIGHT	2	Align right

- int columnNumber: The number of columns. Can use a number between 0 and 30 (0 is the automatic setting)
- int rowNumber: The number of rows. Can use a number between 3 and 90 (0 is the automatic setting)
- int moduleWidth: The width of the module. Can use a number between 1 and 4
- int moduleHeight: The height of the module. Can use a number between 2 and 8
- int eccLevel: The error correction level.

MPosPDF417ECCLevel Enumeration	Value	Description
MPOS_PDF417_ECC_LEVEL_0	48	Errir Correction Level 0
MPOS_PDF417_ECC_LEVEL_1	49	Errir Correction Level 1
MPOS_PDF417_ECC_LEVEL_2	50	Errir Correction Level 2
MPOS_PDF417_ECC_LEVEL_3	51	Errir Correction Level 3
MPOS_PDF417_ECC_LEVEL_4	52	Errir Correction Level 4
MPOS_PDF417_ECC_LEVEL_5	53	Errir Correction Level 5
MPOS_PDF417_ECC_LEVEL_6	54	Errir Correction Level 6
MPOS_PDF417_ECC_LEVEL_7	55	Errir Correction Level 7
MPOS_PDF417_ECC_LEVEL_8	56	Errir Correction Level 8

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_NOT_CONNECT	1006	Failed to connect with the device
MPOS_FAIL_NO_DEVICE	1007	No device connected to B-gate
MPOS_FAIL_INVALID_PARAMETER	1004	Invalid parameter
MPOS_FAIL	1000	Failed

5-2-12 checkPrinterStatus

- Gets the printer's current status.

[Syntax]

Task<uint> checkPrinterStatus(void);

[Returns]

MPosPrinterStatus Enumeration	Value	Description
MPOS_STATUS_NORMAL	0	Printing is possible
MPOS_STATUS_PRINTER_COVEROPEN	1	Cover is open
MPOS_STATUS_PRINTER_PAPEREMPTY	2	No paper
MPOS_STATUS_PRINTER_PAPER_NEAREND	4	Insufficient paper
MPOS_STATUS_PRINTER_ERROR	8	Error

6. MPosLookup Reference

6-1 Overview

• The “**MPosLookup**” interface provides methods getting the connection information of wired/wireless devices and Bluetooth. The connection information from devices can be used for calling “**openService**” method.

6-2 Methods

Note	Use the “ current ” property in “ MPosLookupUtil ” class to call the methods defined in this interface. eg) MPosLookupUtil.Current. refreshDevicesList () eg) MPosLookupUtil.Current. getDeviceList ()
-------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

6-2-1 refreshDevicesList

• Starts searching for Bluetooth devices paired with HOST, and wired/wireless devices on the same network to HOST.

[Syntax]

Task<int> refreshDevicesList (int interfaceType)

[Parameters]

• int interface type: communication type searching for devices.

MPosInterfaceType Enumeration	Value	Description
MPOS_INTERFACE_WIFI	1	WiFi (wireless LAN)
MPOS_INTERFACE_ETHERNET	2	Ethernet (wired LAN)
MPOS_INTERFACE_BLUETOOTH	4	Bluetooth

[Returns]

MPosResult Enumeration	Value	Description
MPOS_SUCCESS	0	Succeeded
MPOS_FAIL_INVALID_INTERFACE	1001	Invalid communication type
MPOS_FAIL_NO_DEVICE_FOUND	5000	No device found

6-2-2 getDevicesList

• Gets the data collections containing the connection information searched by “**refreshDeviceList**” method.

[Parameters]

• int interface type: communication type to get the information.

MPosInterfaceType Enumeration	Value	Description
MPOS_INTERFACE_WIFI	1	WiFi (wireless LAN)
MPOS_INTERFACE_ETHERNET	2	Ethernet (wired LAN)
MPOS_INTERFACE_BLUETOOTH	4	Bluetooth

[Syntax]

ObservableCollection<MposConnectionInformation> getDevicesList (int interfaceType)

[Returns]

•ObservableCollection<MposConnectionInformation>: data collections containing the connection information.

class **MposConnectionInformation**

```
{  
    public MPosInterfaceType IntefaceType { get; set; }  
    public string Name { get; set; }  
    public string Address  
    public string MacAddress { get; set; }  
    public string IpAddress { get; set; }  
    public string PortNumber { get; set; }}
```

Properties

- InterfaceType: indicates one of the communication type supported.
- Name: indicates the display name for the device.
- Address: indicates the address for the device, which used for calling “selectInterface” method.
- MACAddress: indicates MAC address or S/N for Bluetooth device.
- IPAddress: indicates IP address for TCP communication.
- PortNumber: indicates Port number for TCP communication.

Copyright Owned by BIXOLON

This User Manual and product are protected under copyright law.

It is strictly prohibited to copy, duplicate, translate or convert into electronic form the whole or any part of the manual and product without the prior written approval of BIXOLON.

BIXOLON maintains ongoing efforts to enhance and upgrade the functions and quality of all our products. In the following, product specifications and/or user manual content may be changed without prior notice.

The BIXOLON logo is the registered trademark of BIXOLON.

Warning - U.S.A

This equipment has been tested and found to comply with the limits for a Class A digital device pursuant to Part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Notice - Canada

This Apparatus complies with class "A" limits for radio interference as specified in the Canadian department of communications radio interference regulations. Get appareil est conforme aux normes class "A" d'interference radio tel que specifier par ministre canadien des communications dans les reglements d'interference radio.

Caution

Some semiconductor devices are easily damaged by static electricity. You should turn the printer "OFF", before you connect or remove the cables on the rear side, in order to guard the printer against the static electricity. If the printer is damaged by the static electricity, you should turn the printer "OFF".

Copyright © BIXOLON Co., Ltd. 2016-2017. All rights reserved.

Revision history

[illegible]